

Amendments to the Claims

1. (Currently Amended) A method of testing a process that downloads and installs customer ordered software onto a target computer, the method comprising:
 - a. dynamically generating on a simulation computer a file that includes instructions that when executed launch ~~the~~ a process of downloading and ~~the installation of~~ installing on a target computer customer ordered software, including a combination of files from a library, ~~to a target computer~~ that include source code responsible for downloading the customer ordered software onto the target computer;
 - b. simulating execution of said dynamically generated file through recursive evaluation procedures and in accordance with a set of evaluation rules ~~such that to~~ determine which of the combination of files from a library would be executed during a computer manufacturing process and interpret the outcome of the execution of said dynamically generated file ~~is determined;~~
 - c. simulating ~~an~~ a computer manufacturing environment ~~in~~ within which the combination of files from the library run ~~in~~ and interact ~~with~~;
 - d. analyzing the outcome of the simulation of the execution of said dynamically generated file to determine possible syntax errors and possible flow errors.

2. (Original) A method as in claim 1 wherein said dynamically generated file is a main batch file created from a static text file that indicates the model type of the target computer, a lookup file that indicates the necessary instruction required to be executed for the model type indicated, and a process that reads the model type from said static text file and creates said dynamically generated file by reading said lookup file to determine command components.

3. (Original) A method as in claim 2 wherein said main batch file contains one or more labels identifying the flow of the process, one or more commands containing instructions to be executed and one or more calls to one or more static batch files.

4. (Currently Amended) A method as in claim 3 wherein the process of simulating said dynamically generated batch file comprises ~~recursively simulating~~ a recursive

evaluation of each of said one or more batch files to determine the outcome of the process.

5. (Currently Amended) A system of testing a process that downloads and installs customer ordered software onto a target computer, comprising:

- a. a simulation computer ~~comprising an~~ including a simulated computer manufacturing environment, that mimics a target computer;
- b. a first process for creating a second process that launches a process that downloads and installs customer ordered software onto a target computer, including a combination of files from a library, wherein the library includes source code responsible for downloading the customer ordered software onto the target computer ~~simulates the process of downloading and the installation of customer ordered software, including a combination of files from a library, onto the target computer;~~
- c. a third process for examining said combination of files from the library through recursive evaluation procedures and in accordance with a set of evaluation rules to determine which files would be executed in the computer manufacturing environment and interpreting the outcome of simulated execution of the second process; ~~recursively simulating and interpreting the outcome of the execution of the second process; and~~
- d. one or more output files that ~~contain~~ include information relating to the process that downloads and installs customer ordered software onto a target computer. ~~the simulation and interpretation of the second process.~~

6. (Previously Presented) A system as in claim 5 wherein said first process reads a electronic traveler to determine the model of the target computer, looks up in a master token list the model of the target computer and creates from the information in the master token list a second process that is an executable main batch file that downloads and installs customer ordered computer software onto the target computer.

7. (Previously Presented) A system as in claim 6 wherein said main batch file contains labels, commands and sub batch file calls, said third process interpretively tracks said

labels, simulates each of said commands and recursively evaluates each of said sub batch files until the end of the main batch file is reached by said third process.

8. (Cancelled)

9. (Cancelled)

10.(Cancelled)

11.(Cancelled)

12.(Cancelled)

13.(Previously Presented) A method as in claim 1, further comprising reporting said syntax errors and flow errors in a readable format.

14.(Previously Presented) A method as in claim 1, further comprising simulating an entire test process.

15.(Previously Presented) A method as in claim 1, further comprising analyzing an entire simulated test process.

16.(Previously Presented) A method as in claim 1, further comprising simulating a download process.

17.(Previously Presented) A method as in claim 1, further comprising simulating a chip programming process.

18.(Previously Presented) A method as in claim 1, further comprising determining what an outcome of a hypothetical execution would be in a specific simulated environment.

19. (Previously Presented) A method as in claim 1, further comprising taking less than one hour to perform said dynamically generating, said simulating execution, said simulating an environment, and said analyzing the outcome of the simulation.

20. (Previously Presented) A method as in claim 1, further comprising cleaning up errors to facilitate code re-use.

21. (Previously Presented) A system as in claim 5, further comprising an expert tool for facilitating management of software libraries responsible for testing many computers per day.

22. (Previously Presented) A system as in claim 5, wherein the second process facilitates examination of batch files in a simulated environment.

23. (New) A system comprising:

a means for dynamically generating on a simulation computer a file that includes instructions that when executed launch a process of downloading and installing on a target computer customer ordered software, including a combination of files from a library that include source code responsible for downloading the customer ordered software onto the target computer;

a means for simulating execution of said dynamically generated file through recursive evaluation procedures and in accordance with a set of evaluation rules to determine which of the combination of files from a library would be executed during a computer manufacturing process and interpret the outcome of the execution of said dynamically generated file;

a means for simulating a computer manufacturing environment within which the combination of files from the library run and interact;

a means for analyzing the outcome of the simulation of the execution of said dynamically generated file to determine possible syntax errors and possible flow errors.

24. (New) A system as in claim 23 wherein said dynamically generated file is a main batch file created from a static text file that indicates the model type of the target computer, a lookup file that indicates the necessary instruction required to be executed

for the model type indicated, and a process that reads the model type from said static text file and creates said dynamically generated file by reading said lookup file to determine command components.

25.(New) A system as in claim 23 wherein said dynamically generated file includes one or more labels identifying the flow of the process, one or more commands including instructions to be executed and one or more calls to one or more static batch files.